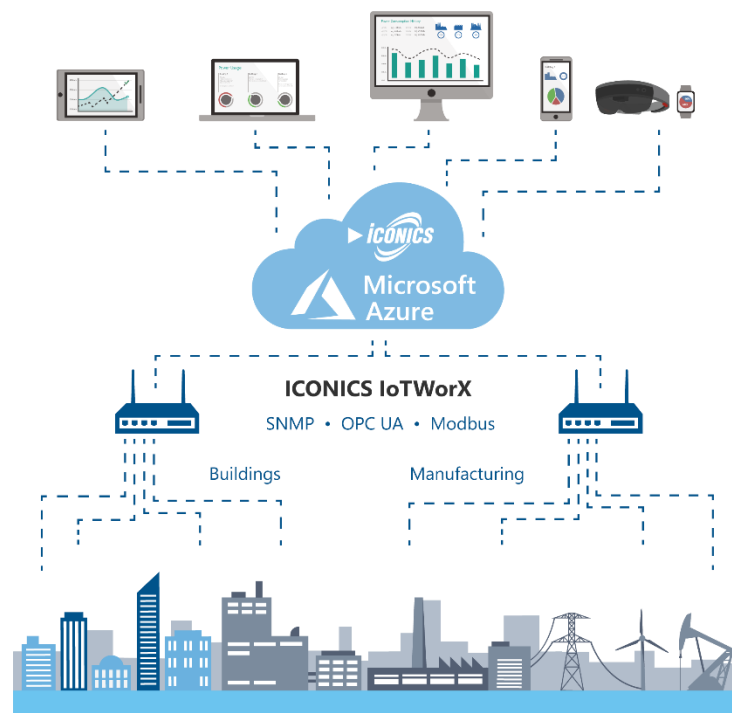


Description: This document describes IoTWorX™ and how to configure it.

Prerequisites:

- Administrative access to Azure Portal
- IoT Hub created
- Azure Storage created



Introduction

IoTWorX bridges the gap between an on-premises communications networks and an Azure cloud-based deployment platform. The IoTWorX device, which can be considered an “edge device”, provides data connectivity between on-premises end devices and the cloud. Targeting process, factory, and building automation as well as other industries, IoTWorX has built-in support for on premise communications including SNMP, Modbus, and OPC UA. The cloud communication path is highly secure and is based on Advanced Message Queuing Protocol (AMQP) and a publish/subscribe (“pub/sub”) mechanism.

Benefits

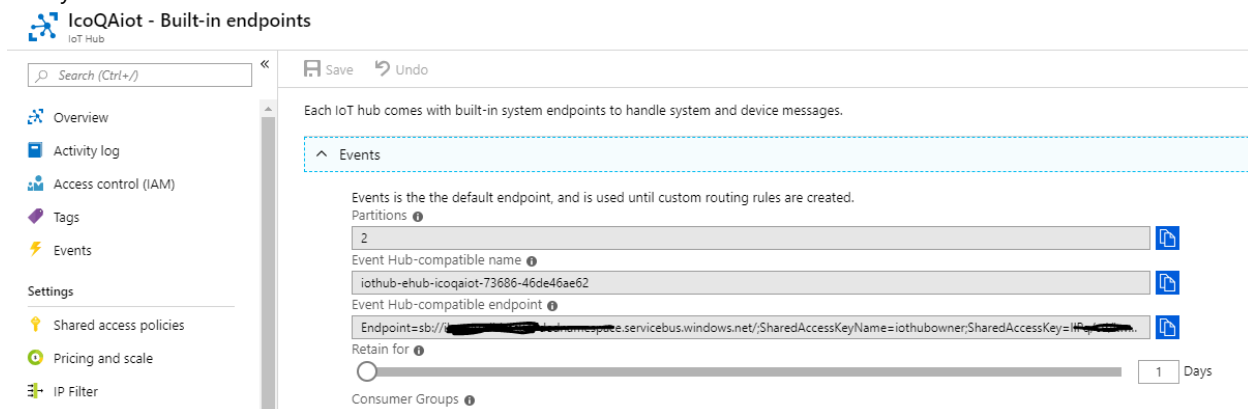
IoTWorX provides manufacturers and facility managers with a flexible platform to create Internet of Things applications. ICONICS offers several key IoT technologies, including rich connectivity to things (OPC, OPC UA, SNMP, Modbus), secure cloud communications, and built-in real-time visualization and analytics. With simple setup and configuration, users can easily create remote monitoring and analytics solutions that meet

their innovative business requirements for collaboration and shared insight across widely dispersed assets. The ICONICS IoT solution leverages the Azure cloud to provide global visibility, scalability, and reliability. Connect to standard ICONICS applications in the cloud such as GENESIS64™, Hyper Historian™, AnalytiX®, and more. Optionally integrate Microsoft Azure services such as Power BI and Machine Learning to provide greater depth of analyses.

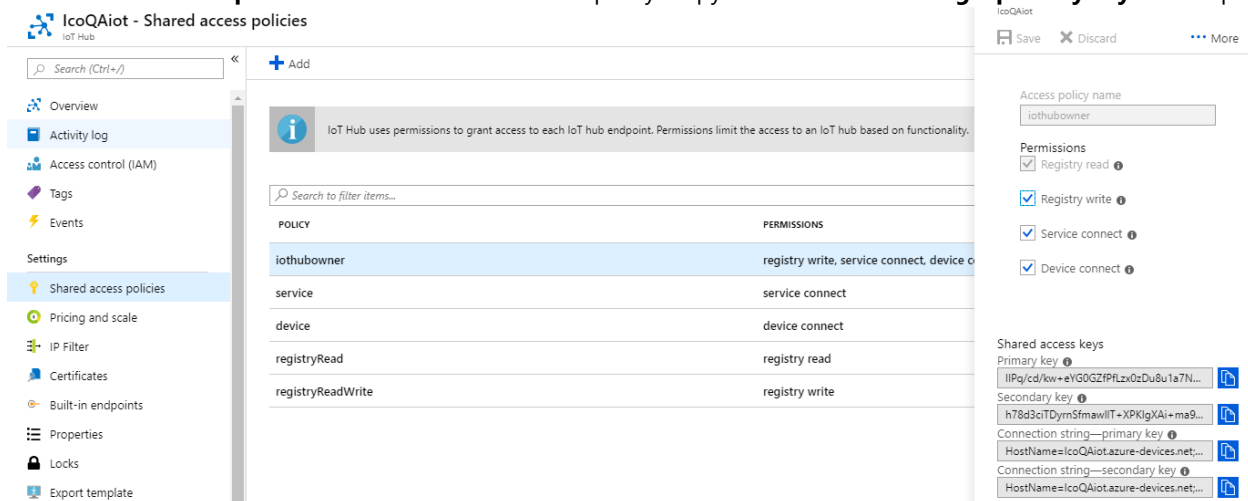
Creating and Gathering Azure Resources

- 1) In the Azure portal, select your Azure storage account and then **Access Keys**. Copy the **Connection string – primary key** and keep it handy – this will be used in a later step.

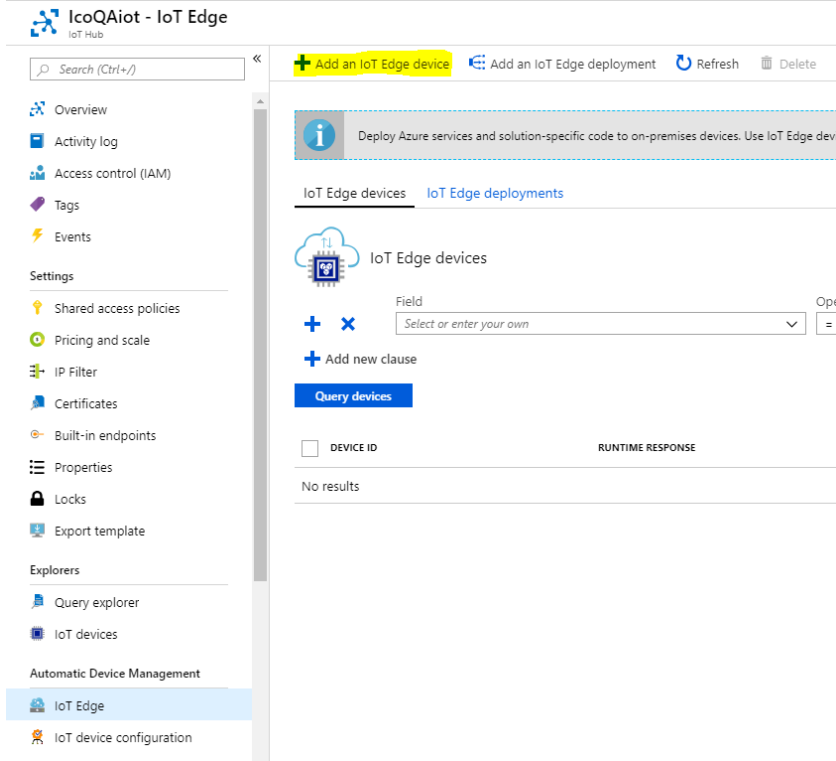
- 2) In the Azure portal, select your IoT Hub, select **Built-in endpoints**. Copy the **Event Hub-compatible endpoint** and keep it handy.



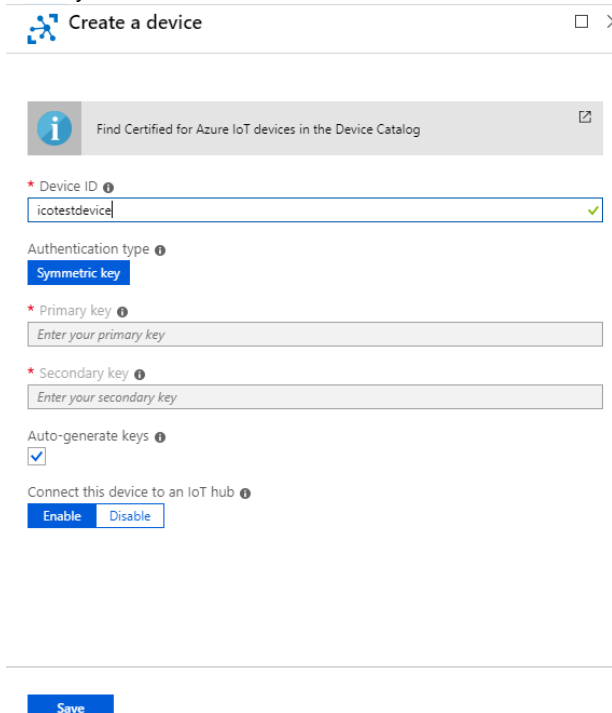
- 3) Go to **Shared access policies**. Select the **iothubowner** policy. Copy the **Connection string – primary key** and keep it handy.



- 4) Now we are ready to create a device. Still on the **IoT Hub** page, select **IoT Edge** and then **Add an IoT Edge device**.



- 5) Name your device and then click **Save**.



6) The device will now be listed under **IoT Edge Devices**.

[+ Add an IoT Edge device](#)
[+ Add an IoT Edge deployment](#)
[Refresh](#)
[Delete](#)


 Deploy Azure services and solution-specific code to on-premises devices. Use IoT Edge devices to perform compute and analytics tasks on data before it's sent to the cloud.

[IoT Edge devices](#)
[IoT Edge deployments](#)


 IoT Edge devices

[+ Add new clause](#)

Field:
 Operator:
 Value:

[Query devices](#)

DEVICE ID	RUNTIME RESPONSE	IOT EDGE MODULE COUNT	CONNECTED CLIENT COUNT
icotestdevice	N/A	0	0

7) Click the device to be brought to the overview page.

[Device details](#)
[icotestdevice](#)

[Save](#)
[Set modules](#)
[Device twin](#)
[Regenerate keys](#)
[Refresh](#)

Device ID:

Primary key:

Secondary key:

Connection string (primary key):

Connection string (secondary key):

Connect this device to an IoT hub: ☒ Enable ☐ Disable

Edge runtime response:

[Modules](#)
[IoT Edge hub connections](#)
[Deployments](#)


 Verify that your modules are included in the deployment, and whether your modules have been reported by the device. Click Set modules to change the modules that appear. Each device can host a maximum of 20 modules.

NAME	TYPE	SPECIFIED IN DEPLOYMENT	REPORTED BY DEVICE	RUNTIME STATUS	EXIT CODE
\$edgeAgent	Module Identity	N/A	N/A	N/A	N/A
\$edgeHub	Module Identity	N/A	N/A	N/A	N/A

8) Copy the **Connection String** and keep handy.

Setting Up the IoTWorX Edge Device

Install IoTWorX on Ubuntu Edge Device

If you are using Ubuntu 16.04 or Ubuntu 18.04 (server or desktop version), you can use the IoTWorX installation to prepare the device.

- 1) Cd into the **Linux64/Disk1/InstData/VM** folder.
- 2) Run this command: **sudo sh ./IcoIoT_64.bin**
- 3) When the installation is finished, proceed to the Creating an IoT Project section.

Install IoTWorX on Other Edge Devices

For other (non-Ubuntu) operating systems, including ARM32 based systems, use the following steps to prepare the device:

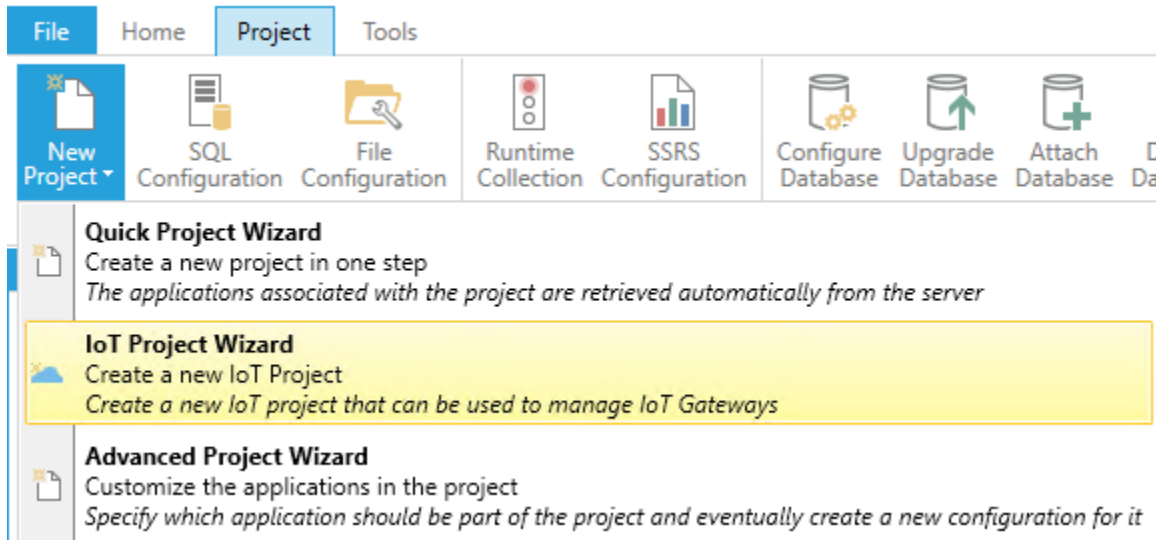
- 1) On your IoT device, install **Azure IoT Edge runtime**. Directions for doing so can be found here:
 - a. For x64 devices - <https://docs.microsoft.com/en-us/azure/iot-edge/how-to-install-iot-edge-linux>
 - b. For ARM32 devices - <https://docs.microsoft.com/en-us/azure/iot-edge/how-to-install-iot-edge-linux-arm>

Note: Use the Edge Device connection string obtained in step 8 of the previous section when editing the `/etc/iotedge/config.yaml` file. For production systems, you should also install certificates for IoT Edge. More information can be found here: <https://docs.microsoft.com/en-us/azure/iot-edge/how-to-create-transparent-gateway>

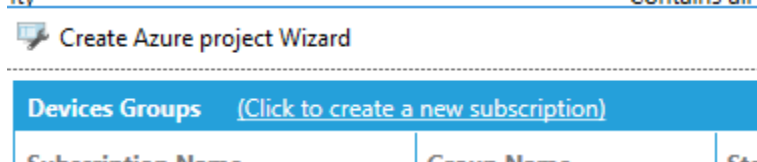
- 2) If you run **sudo iotedge list** on the device, you will see that the edgeAgent module is running.
- 3) Run **sudo mkdir /usr/share/ICONICS** to create a directory where some common program data for ICONICS modules will be stored.
- 4) Copy the contents of the provided zip file into **/usr/share/ICONICS**.
- 5) Run **sudo chmod 755 /usr/share/ICONICS** to enable the required permissions for all files.
- 6) For production environments, update the Docker log file settings to limit the size (see: <https://docs.microsoft.com/en-us/azure/iot-edge/production-checklist>).

Creating an IoT Project

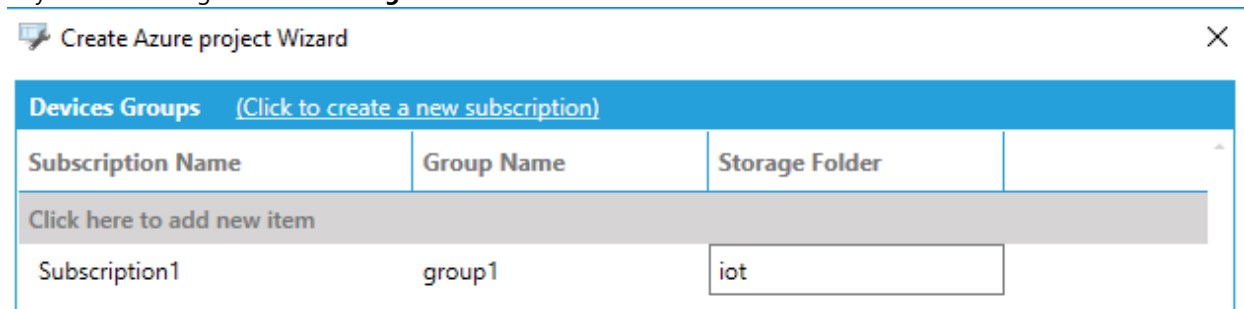
- 1) On the server where ICONICS Suite is installed, open **Workbench** and launch the **IoT Project Wizard**.



- 2) Give your project some name. In the **Storage Connection** field, enter the connection string obtained in the first step of this document, then click **Next**.
- 3) Click the **Click to create a new subscription** link.

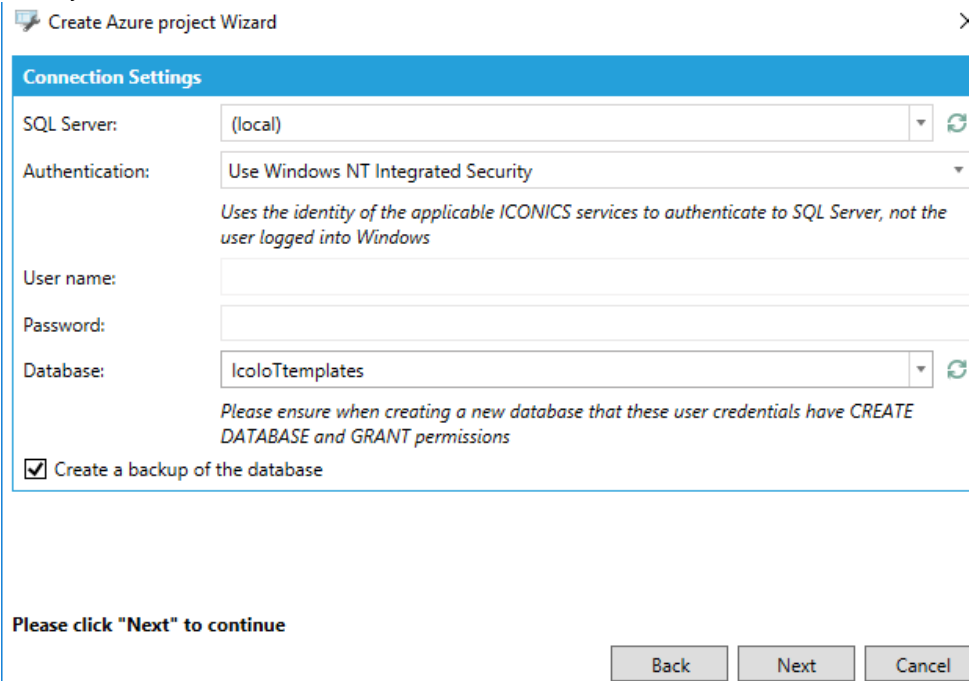


- 4) Enter your subscription, and then enter the Event Hub connection string and IoT Hub owner connection string obtained in step 2 and step 3 of the first section of this document. Click **OK**.
- 5) Create a new entry in the device groups list with the subscription you just created. Give the group a name, then enter the name of your blob storage folder in **Storage Folder**.



- 6) Click **Next**.

- 7) Select your SQL Server, then enter the name of a NEW database where we will store the template configuration for our devices.



Create Azure project Wizard

Connection Settings

SQL Server: (local) 

Authentication: Use Windows NT Integrated Security 

Uses the identity of the applicable ICONICS services to authenticate to SQL Server, not the user logged into Windows

User name:

Password:

Database: IcoloTemplates 

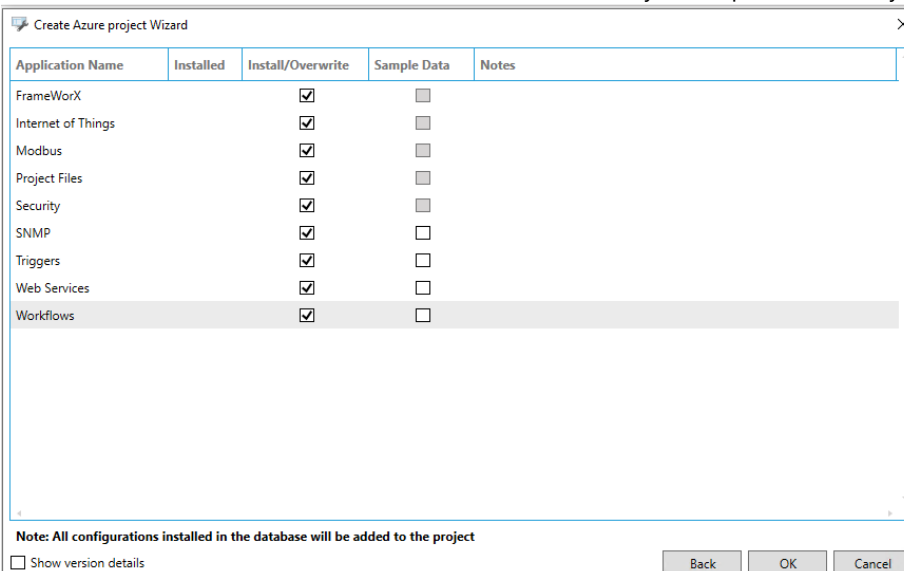
Please ensure when creating a new database that these user credentials have CREATE DATABASE and GRANT permissions

☒ Create a backup of the database

Please click "Next" to continue

Back Next Cancel

- 8) Click **Next**, then **Next** again.
- 9) Check the **Install/Overwrite** checkbox for each module that you will plan to use on your IoT device.



Create Azure project Wizard

Application Name	Installed	Install/Overwrite	Sample Data	Notes
FrameWorX	☒	☒	☐	
Internet of Things	☒	☒	☐	
Modbus	☒	☒	☐	
Project Files	☒	☒	☐	
Security	☒	☒	☐	
SNMP	☒	☒	☐	
Triggers	☒	☒	☐	
Web Services	☒	☒	☐	
Workflows	☒	☒	☐	

Note: All configurations installed in the database will be added to the project

☐ Show version details

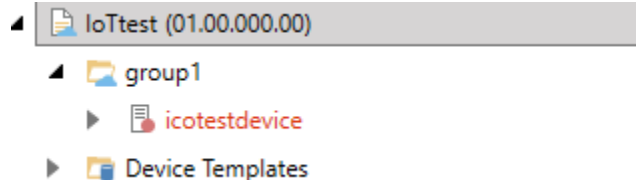
Back OK Cancel

Note: The sample data for several modules could be optionally generated in the new database. Check the **Sample Data** checkbox for each module that you want to generate them.

- 10) Click **OK** to create the project.

Deploying ICONICS Modules to the Device

- 1) Expand the new IoT project in **Workbench** to see the device listed:



- 2) The device appears red here to indicate we can't deploy new ICONICS configurations to the device yet. We first need ICONICS modules running on the device before we can update its configuration. To do this, right click on device and select **Configure Modules**.
- 3) On the next screen, you will see the status of any Edge modules running on the device. Click the **Click to configure modules** link in the **Available Modules** header.

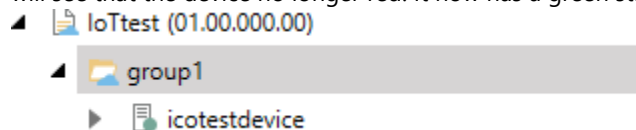
Configure Modules for IoT Edge Device ×

Available Modules (Click to configure modules)				
Name	Device	Type	Status	Version
\$edgeAgent	icotestdevice	System	Running	1.0.6
\$edgeHub	icotestdevice	System	Running	1.0.6

- 4) Under the **Deployment Options** section, selection an option from the **Version** and **Architecture** dropdowns that are applicable to your deployment.
- 5) Check the **Deploy** checkbox for each ICONICS module you would like to add to your IoT device.

Note: The minimum required modules are **FrameWorX Server**, **IoT Publisher** and **Ico Edge Agent**. As we are using the IoT Visualizer for local visualizations later in this document we recommend deploying also IoT Visualizer module to IoT device already in this step.

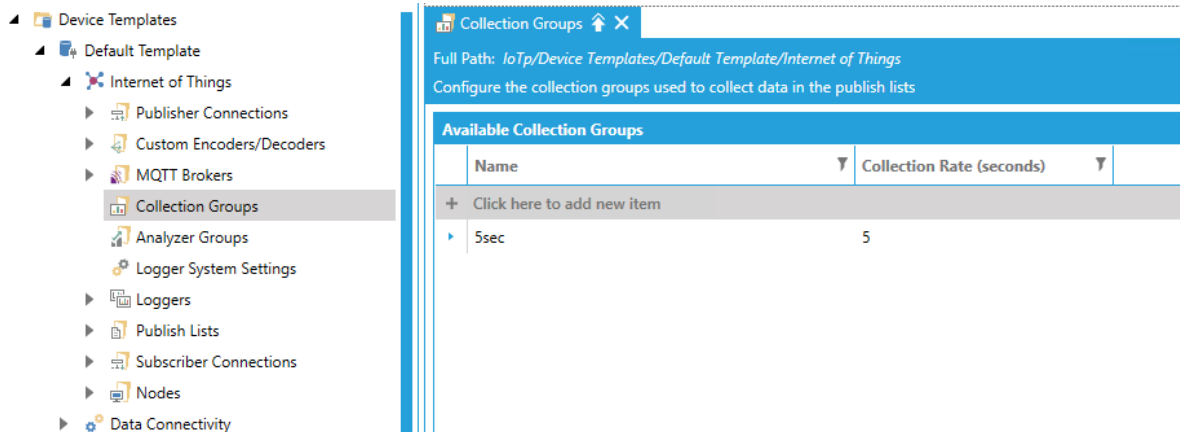
- 6) Click **Deploy**. You should see a **Deploy successfully completed** message, indicating that a new deployment has been queued for the device. Click **Close**.
- 7) At this point, ICONICS modules will begin downloading to the IoT device. Depending on the speed of your internet, this may take some time (ICONICS modules total about 1.6 GB).
- 8) After modules have finished downloading and have started on the device, if you refresh your device group in **Workbench**, you will see that the device no longer red. It now has a green status icon indicating that it is online and running.



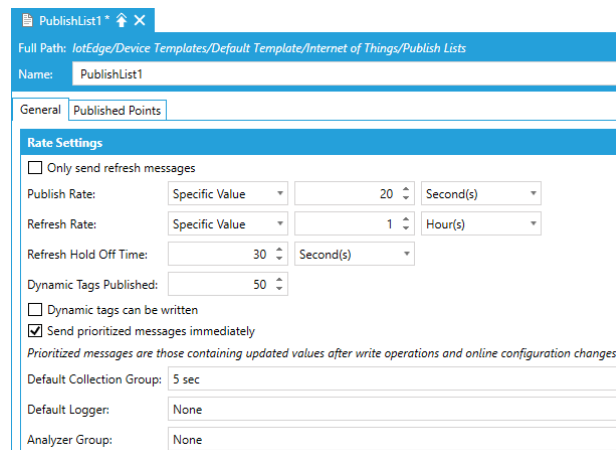
Deploying Configurations to the Device

- 1) Now that ICONICS modules are running on the gateway, we can deploy configurations to it. First, we will define a collection group. Expand your **Default Template** and **Internet of Things** provider. Right click **Collection Groups** and select **Edit**.

- 2) Add a new **Collection Group** to the list. This defines how often tags will be collected on the device. Apply your changes and close the form.

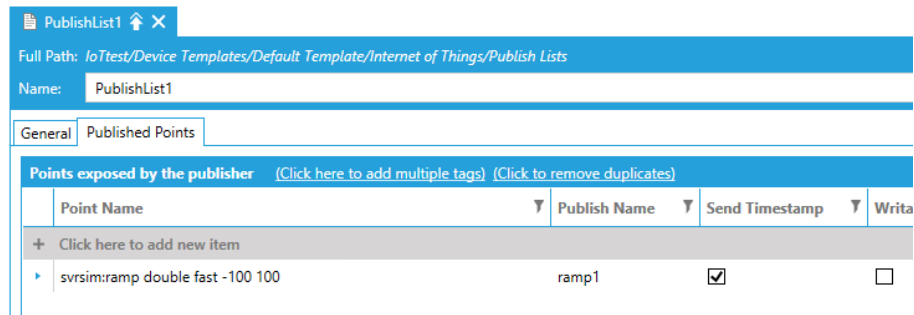


- 3) Next, we will define a publish list. Right click **Publish Lists** and select **Add Publish List**. Give your list a name and change the **Default Collection Group** to the group created in the previous step. Change the **Publish Rate** and **Refresh Rate** to values appropriate for your deployment.



Note: There is an upper limit of 50 tags in dynamic subscriptions. Users are strongly encouraged to use static publish lists for this reason. Dynamic subscriptions are intended for small ad hoc requests for data, not large sets of tags.

- 4) On the **Published Points** tab, add the points you would like to be published. As an example, you can enter **svrsim:ramp double fast -100 100** as a point name with a **Publish Name** of **ramp1**. Apply your changes and close the form.



Full Path: *IoTTest/Device Templates/Default Template/Internet of Things/Publish Lists*

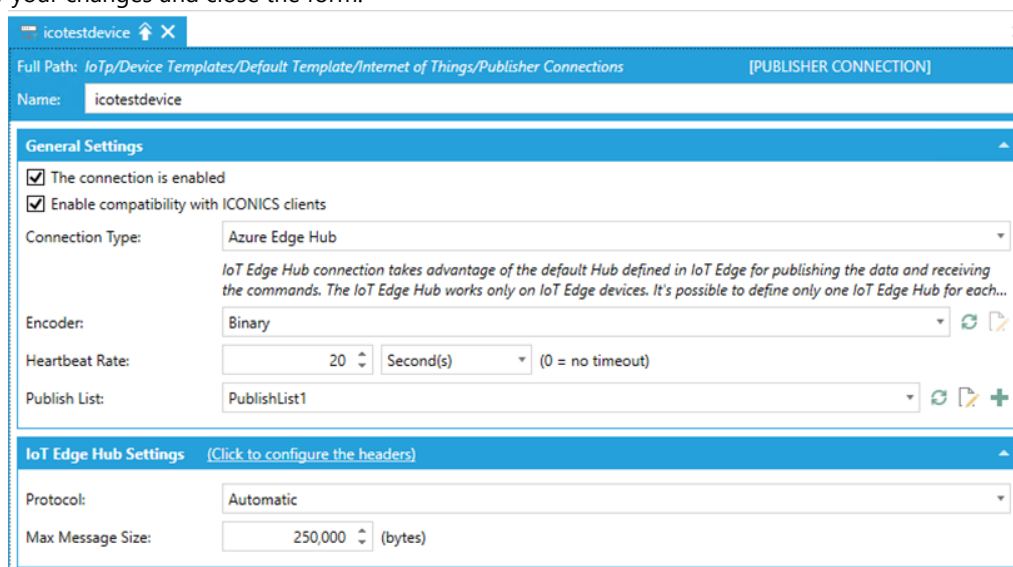
Name: PublishList1

General Published Points

Points exposed by the publisher [\(Click here to add multiple tags\)](#) [\(Click to remove duplicates\)](#)

Point Name	Publish Name	Send Timestamp	Write
+ Click here to add new item			
svrsim:ramp double fast -100 100	ramp1	☒	☐

- 5) Open the **Publisher Connection** form for your device. Specify the publish list you created in the previous step as the **Publish List**. Apply your changes and close the form.



Full Path: *IoT/Device Templates/Default Template/Internet of Things/Publisher Connections* [PUBLISHER CONNECTION]

Name: icotestdevice

General Settings

☒ The connection is enabled

☒ Enable compatibility with ICONICS clients

Connection Type: Azure Edge Hub

IoT Edge Hub connection takes advantage of the default Hub defined in IoT Edge for publishing the data and receiving the commands. The IoT Edge Hub works only on IoT Edge devices. It's possible to define only one IoT Edge Hub for each...

Encoder: Binary

Heartbeat Rate: 20 Second(s) (0 = no timeout)

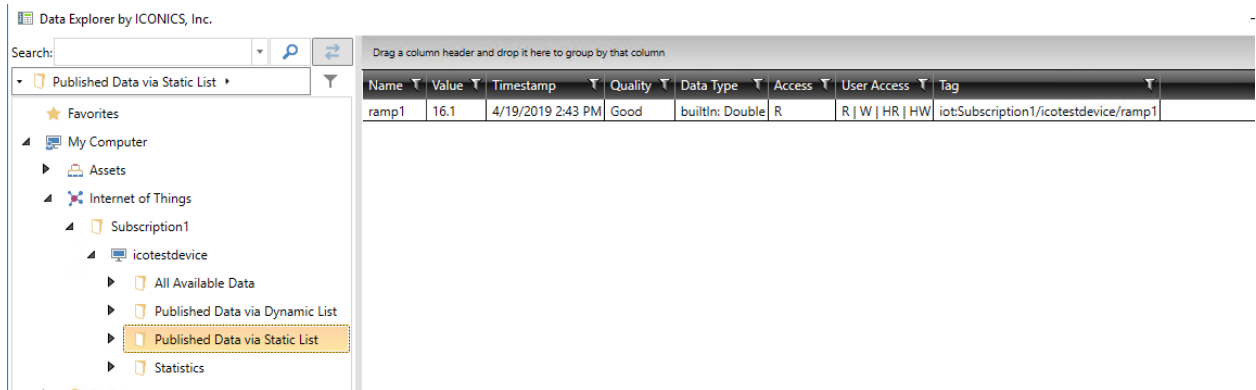
Publish List: PublishList1

IoT Edge Hub Settings [\(Click to configure the headers\)](#)

Protocol: Automatic

Max Message Size: 250,000 (bytes)

- 6) Now we will deploy this configuration to the device. Right click your device and select **Deploy Device(s) Configuration**. **Workbench** will create a new task to accomplish this. What is happening under the hood is:
 - a. The configuration that has been created in Workbench and stored in SQL Server is exported and a SQLite file is created.
 - b. The SQLite file (and other configuration pieces such as custom KPI dashboards) are packaged and sent to the Azure Storage account.
 - c. The Edge device's module twin is updated with the location of the new configuration.
 - d. The device receives a notification that its twin was updated and will download and apply the new configuration.
- 7) To confirm the new configuration has been applied, open **Data Explorer**. Browse to your device and look at the **Published Data via Static List**. You should see the point(s) you had defined earlier.



The screenshot shows the 'Data Explorer by ICONICS, Inc.' interface. On the left is a tree view under 'Published Data via Static List' with the following structure:

- My Computer
 - Assets
 - Internet of Things
 - Subscription1
 - icotestdevice
 - All Available Data
 - Published Data via Dynamic List
 - Published Data via Static List (highlighted)
 - Statistics

On the right is a data table with the following columns: Name, Value, Timestamp, Quality, Data Type, Access, User Access, and Tag. The table contains one row of data:

Name	Value	Timestamp	Quality	Data Type	Access	User Access	Tag
ramp1	16.1	4/19/2019 2:43 PM	Good	builtin: Double	R	R W HR HW	iot:Subscription1/icotestdevice/ramp1

Set Diagnostics and Configuration Admin Password

It's highly recommended to change the default password for the Diagnostics and Configuration pages.

- 1) Go to the **IoT WorX Landing Page** by browsing to <http://DeviceIPaddress>.
- 2) Select **Diagnostics and Configuration**.
- 3) Log in as **admin** with the default password of **iconics**.
- 4) Follow the prompts to set a new admin password.